Group Latent Factor Model for Recommendation with Multiple User Behaviors

Jian Cheng, Ting Yuan, Jinqiao Wang, Hanqing Lu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{jcheng, tyuan, jqwang, luhq}@nlpr.ia.ac.cn

ABSTRACT

Recently, some recommendation methods try to relieve the data sparsity problem of Collaborative Filtering by exploiting data from users' multiple types of behaviors. However, most of the exist methods mainly consider to model the correlation between different behaviors and ignore the heterogeneity of them, which may make improper information transferred and harm the recommendation results. To address this problem, we propose a novel recommendation model, named Group Latent Factor Model (GLFM), which attempts to learn a factorization of latent factor space into subspaces that are shared across multiple behaviors and subspaces that are specific to each type of behaviors. Thus, the correlation and heterogeneity of multiple behaviors can be modeled by these shared and specific latent factors. Experiments on the real-world dataset demonstrate that our model can integrate users' multiple types of behaviors into recommendation better.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Recommender Systems; Matrix Factorization

1. INTRODUCTION

In the past decade, Collaborative Filtering (CF) has become one of the most popular techniques for recommender systems, which makes predictions by mining users' historical behaviors on items. In particular, Matrix Factorization (MF) models [6, 2] have become dominant in current CF methods. MF methods learn low-dimensional latent factor vectors of users and items to represent their characteristics, and predictions are made by the inner product of them. Traditionally, CF methods are designed to deal with single type of user behavior at a time. However, the behavioral data is typically very sparse, that is, most users have interacted with a very few items. It is indeed hard for CF methods including MF to make accurate recommendations with such insufficient data.

To address the data sparsity problems, some works have been proposed to exploit users' multiple types of behaviors for recommendation [1, 7, 4]. As we know, with the prevalence of massive web applications, users often have various types of behaviors on the web, varying from rating movies, listening music, to making friends. Considering simultaneously multiple behaviors of a user offers us more information to model the user's taste better. The most widely used method addressing this issue is Collective Matrix Factorization (CMF) [7], which decomposes the rating matrices for different types of user behaviors jointly by sharing the same user latent factor matrix across different behaviors. That is, in CMF a user is characterized by the same latent factor vector across different behaviors. Through these shared user latent factors, CMF aims to transfer information between different behaviors to improve the recommendation results. Inspired by the idea of CMF, some following works [5, 8, 3] have demonstrated that better predictions can be achieved by sharing the same user latent factors across multiple types of behaviors.

However, the CMF ignores the heterogeneity of different behaviors. When characterizing a user by the same latent factor vector across different behaviors, the underlying assumption is that user's taste should be the same when she/he conducts different behaviors. This is too strict to be realistic. For example, a user's taste on music may be quite different from her/his taste on movie. When users conduct different types of behaviors, there should exist some correlations between behaviors as well as some specific characteristics for each type of behaviors. Nevertheless, traditional methods like CMF mainly consider to model the correlation of different behaviors but neglect the heterogeneity of them, which could not effectively model the characteristics of a user on various behaviors and may make improper information transferred to harm the recommendation results.

In this paper, we propose to integrate multiple types of user behaviors into recommendation effectively by modeling both the correlation and heterogeneity of them. Particularly, we present a novel recommendation model, named Group Latent Factor Model (GLFM), which attempts to learn a factorization of the latent factor space into subspaces that are shared across multiple behaviors and subspaces that are specific to each type of behaviors. In GLFM, when user conducts different types of behaviors, she/he is character-

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ized by different latent factor vectors, among which, some dimensions are shared by multiple types of behaviors while the others are specific for certain behavior. Thus, the correlation and heterogeneity of multiple behaviors can be modeled by these shared and specific latent factors. Experiments show that our method can achieve better recommendation results than other state-of-the-art methods.

2. THE PROPOSED METHOD

2.1 Problem Statement

Suppose that we have a set of n users $\mathcal{U} = \{u_1, \cdots, u_n\}$ and their B types of behavior records. Each type of user behavior demonstrating her/his opinions on a kind of items can be regarded as ratings (binary or real values), thus we have B rating matrices for different behaviors, denoted as $\{\mathbf{R}^1, \cdots, \mathbf{R}^B\}$, where $\mathbf{R}^b = [R^b_{ij}]_{n \times m_b}$ denotes the rating matrix for the b-th type of behavior. R^b_{ij} denotes the rating of u_i on item v^b_j, v^b_j denotes the j-th item in the b-th type of behavior, m_b is the number of items belong to the b-th type. Our goal is to predict the missing values in each behavior matrix $\mathbf{R}^b$ ($b = 1, \cdots, B$) by effectively exploiting the observed data from users' multiple types of behaviors.

2.2 Group Latent Factor Model

We formulate our problem on the basis of Matrix Factorization [6], which learns latent factors of the users and the items to characterize them. In our cases, for each user we have her/his multiple types of behavior records, leading to multiple rating matrices with the same user dimension. To correctly account for the correlation and heterogeneity of different behaviors, we cast the problem as finding a factorization of the latent factor space into subspaces that are shared across multiple behaviors and subspaces that are specific to each type of behaviors.

Let $\mathbf{U}^0 \in \mathbb{R}^{K_s \times n}$ denote the user latent factor matrix shared among different behaviors, with each column U^0_i representing the shared latent factor vector for user u_i. K_s is the number of the shared factors. For the b-th type of behaviors, let $\mathbf{U}^b \in \mathbb{R}^{K_b \times n}$ denote the behavior-specific user latent factor matrix, with each column U^b_i representing the behavior-specific latent factor vector for user u_i. K_b is the number of the specific factors for the b-th type of behaviors. As shown in Figure 1, when user u_i conducts the b-th behaviors, she/he is modeled by $\tilde{U}^b_i = [U^0_i; U^b_i] \in \mathbb{R}^{(K_s+K_b)\times 1}$, which consists of both the shared and the behavior-specific latent factors. We denote $\tilde{\mathbf{U}}^b \in \mathbb{R}^{(K_s+K_b)\times n}$ to be the b-th user latent factor matrix with each column as $\tilde{U}^b_i$, thus we have $\tilde{\mathbf{U}}^b = [\mathbf{U}^0; \mathbf{U}^b]$.

For items belong to the b-th type, we let $\mathbf{V}^b \in \mathbb{R}^{(K_s+K_b)\times m_b}$ denote the item latent factor matrix, with each column V^b_j representing the latent factor vector for item v^b_j. We denote $V^b_j = [D^b_j; P^b_j]$, where $D^b_j \in \mathbb{R}^{K_s \times 1}$ corresponds to the shared latent factor space of U^0_i and $P^b_j \in \mathbb{R}^{K_b \times 1}$ corresponds to the behavior-specific latent factor space of U^b_i. Thus, the ratings of user u_i on item v^b_j can be predicted as:

$$\hat{R}^b_{ij} = (\tilde{U}^b_i)^T V^b_j = [U^0_i; U^b_i]^T V^b_j = (U^0_i)^T D^b_j + (U^b_i)^T P^b_j \quad (1)$$

Thus, given users' B types of behavior records $\{\mathbf{R}^1, \cdots, \mathbf{R}^B\}$, we learn $\{\tilde{\mathbf{U}}^b\}^B_{b=1}$ and $\{\mathbf{V}^b\}^B_{b=1}$ by minimizing the following objective function:

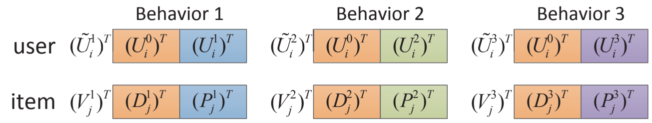

Figure 1: In GLFM, user u_i is modeled by $\tilde{U}^b_i = [U^0_i; U^b_i]$ when she/he conducts the b-th behavior, where U^0_i is shared across multiple behaviors and U^b_i is specific for the b-th behavior. v^b_j, the j-th item belongs to the b-th type, is modeled by $V^b_j = [D^b_j; P^b_j]$, where D^b_j corresponds to the latent factor space of U^0_i and P^b_j corresponds to that of U^b_i.

$$\begin{aligned}
&\mathcal{L}(\{\tilde{\mathbf{U}}^b\}^B_{b=1}, \{\mathbf{V}^b\}^B_{b=1}) \\
&= \sum_{b=1}^{B} \left(\sum_{i=1}^{n} \sum_{j=1}^{m_b} I^b_{ij} (R^b_{ij} - (\tilde{U}^b_i)^T V^b_j)^2 + \lambda (\|\tilde{\mathbf{U}}^b\|^2_F + \|\mathbf{V}^b\|^2_F) \right) \\
&= \sum_{b=1}^{B} \sum_{i=1}^{n} \sum_{j=1}^{m_b} I^b_{ij} \left(R^b_{ij} - (U^0_i)^T D^b_j - (U^b_i)^T P^b_j \right)^2 \\
&\quad + \lambda (B \sum_{i=1}^{n} \|U^0_i\|^2_F + \sum_{b=1}^{B} \sum_{i=1}^{n} \|U^b_i\|^2_F + \sum_{b=1}^{B} \sum_{j=1}^{m_b} \|V^b_j\|^2_F)
\end{aligned} \quad (2)$$

In Eq.(2), the first term measures the quality of the approximation of the predicted ratings to the observed ratings in multiple types of user behaviors by squared error, where I^b_{ij} is the indicator function which is equal to 1 if the user u_i rated the item v^b_j and is 0 otherwise. In the second term, $\|\cdot\|_F$ is the Frobenius regularization norm which is used to avoid over-fitting. Parameter λ controls the strength of the regularization term.

Notice that, traditional MF can be viewed as a special case of our model by restricting the number of shared latent factors $K_s = 0$; CMF can be viewed as a special case of our model by restricting the number of behavior-specific latent factors $K_b = 0$.

2.3 Optimization Algorithm

Eq.(2) is convex with respect to one of the variables $\mathbf{U}^0$, $\mathbf{U}^1, \cdots, \mathbf{U}^B, \mathbf{V}^1, \cdots, \mathbf{V}^B$ when the others are fixed. Thus, we apply an alternating optimization to solve the problem, which update $\mathbf{U}^0$, $\{\mathbf{U}^b\}^B_{b=1}$, and $\{\mathbf{V}^b\}^B_{b=1}$ iteratively and alternatingly.

Optimizing $\mathbf{U}^0$, when $\{\mathbf{U}^b\}^B_{b=1}$ and $\{\mathbf{V}^b\}^B_{b=1}$ fixed: $\mathbf{U}^0$ can be obtained by solving following optimization problem,

$$\begin{aligned}
\min_{\mathbf{U}^0} \mathcal{L}(\mathbf{U}^0) &= \sum_{b=1}^{B} \sum_{i=1}^{n} \sum_{j=1}^{m_b} I^b_{ij} \left(R^b_{ij} - (U^0_i)^T D^b_j - (U^b_i)^T P^b_j \right)^2 \\
&\quad + \lambda B \sum_{i=1}^{n} \|U^0_i\|^2_F
\end{aligned} \quad (3)$$

solving $\frac{\partial \mathcal{L}(\mathbf{U}^0)}{\partial U^0_i} = 0$, we have:

$$\begin{aligned}
U^0_i &= \left(\lambda B \mathbf{E}_0 + \sum_{b=1}^{B} \sum_{j=1}^{m_b} I^b_{ij} D^b_j (D^b_j)^T \right)^{-1} \\
&\quad \times \left(\sum_{b=1}^{B} \sum_{j=1}^{m_b} I^b_{ij} (R^b_{ij} - (U^b_i)^T P^b_j) D^b_j \right)
\end{aligned} \quad (4)$$

where $\mathbf{E}_0$ is a $K_s \times K_s$ identity matrix.

Algorithm 1 Optimization Algorithm for GLFM

Require: $\{\mathbf{R}^b\}_{b=1}^B$, Parameters K_s, K_b, λ
Ensure: $\mathbf{U}^0$, $\{\mathbf{U}^b\}_{b=1}^B$ and $\{\mathbf{V}^b\}_{b=1}^B$

```
Initialize U^0, {U^b}_{b=1}^B and {V^b}_{b=1}^B;
Repeat
  Update U_i^0, ∀1 ≤ i ≤ n with Eq. (4)
  for b = 1 to B do
    Update U_i^b, ∀1 ≤ i ≤ n with Eq. (6)
  end for
  for b = 1 to B do
    Update V_j^b, ∀1 ≤ j ≤ m_b with Eq. (8);
  end for
Until convergence
Return U^0, {U^b}_{b=1}^B and {V^b}_{b=1}^B
```

Optimizing $\{\mathbf{U}^b\}_{b=1}^B$, given $\mathbf{U}^0$ and $\{\mathbf{V}^b\}_{b=1}^B$: When $\mathbf{U}^0$ and $\{\mathbf{V}^b\}_{b=1}^B$ are fixed, the constraints are independent on each $\mathbf{U}^b$ ($b = 1, \cdots, B$), suggesting that we can optimize each $\mathbf{U}^b$ separately. $\mathbf{U}^b$ can be obtained by solving following problem,

$$\min_{\mathbf{U}^b} \mathcal{L}(\mathbf{U}^b) = \sum_{i=1}^{n}\sum_{j=1}^{m_b} I_{ij}^b \left(R_{ij}^b - (U_i^0)^T D_j^b - (U_i^b)^T P_j^b\right)^2 + \lambda \sum_{i=1}^{n} \|U_i^b\|_F^2 \tag{5}$$

solving $\frac{\partial \mathcal{L}(\mathbf{U}^b)}{\partial U_i^b} = 0$, we have:

$$U_i^b = \left(\lambda \mathbf{E}_b + \sum_{j=1}^{m_b} I_{ij}^b P_j^b (P_j^b)^T\right)^{-1} \left(\sum_{j=1}^{m_b} I_{ij}^b (R_{ij}^b - (U_i^0)^T D_j^b) P_j^b\right) \tag{6}$$

where $\mathbf{E}_b$ is a $K_b \times K_b$ identity matrix.

Optimizing $\{\mathbf{V}^b\}_{b=1}^B$, given $\mathbf{U}^0$ and $\{\mathbf{U}^b\}_{b=1}^B$: We can also optimize each $\mathbf{V}^b$ separately by solving following problem:

$$\min_{\mathbf{V}^b} \mathcal{L}(\mathbf{V}^b) = \sum_{i=1}^{n}\sum_{j=1}^{m_b} I_{ij}^b \left(R_{ij}^b - (\tilde{U}_i^b)^T V_j^b\right)^2 + \lambda \sum_{j=1}^{m_b} \|V_j^b\|_F^2 \tag{7}$$

solving $\frac{\partial \mathcal{L}(\mathbf{V}^b)}{\partial V_j^b} = 0$, we have:

$$V_j^b = \left(\lambda \mathbf{E}_{sb} + \sum_{i=1}^{n} I_{ij}^b \tilde{U}_i^b (\tilde{U}_i^b)^T\right)^{-1} \left(\sum_{i=1}^{n} I_{ij}^b R_{ij}^b \tilde{U}_i^b\right) \tag{8}$$

where $\mathbf{E}_{sb}$ is a $(K_s + K_b) \times (K_s + K_b)$ identity matrix.

The detailed optimization algorithm is described in Algorithm 1. Note that, the number of behavior-specific latent factors K_b can be different for different behaviors. In order to reduce the model complexity, we set all the K_b ($b = 1, \cdots, B$) the same in our experiments.

3. EXPERIMENTS

3.1 Experiment Settings

Datasets: To evaluate our model's recommendation quality, we crawled the dataset from the publicly available website Douban[1], where users can provide their ratings for movie, books and music, as well as establish social relations with others. Thus, we have four types of user behaviors here. To have sufficient observations to be split in various proportions of training and testing data for our evaluation, we filtered out users who have rated less than 10 books, or 10 movie, or 10 music, and then removed users without social relationships with others. Retrieving all items rated by the selected users, we have a dataset containing 5,916 users with their ratings on 14,155 books, 15,492 music and 7,845 movie, as well as their social relations between each other. The ratings are real values in the range [1,5], while the social relations are binary, indicating whether or not a social relation exists. The detailed statistics are showed in Table 1.

Table 1: Statistics of the Datasets

Behavior Type	#Items	Sparsity	#Ratings per User
Book	14,155	99.85%	22
Music	15,492	99.75%	38
Movie	7,845	98.87%	88
Social Relation	5,916	99.72%	17

Performance Metric: We focus on the task of rating prediction in recommendation to evaluate our models' quality. The most popular metric, Root Mean Square Error (RMSE) is used to measure the prediction quality.

$$RMSE = \sqrt{\frac{1}{T}\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2} \tag{9}$$

where R_{ij} and $\hat{R}_{ij}$ denote the true and predicted ratings respectively, and T denotes the number of tested ratings. The smaller RSME value means a better performance.

Baseline Methods: For comparison, we consider following related methods: (1)PMF [6], the state-of-the-art traditional MF method, which learns latent factors for each type of behaviors separately with no information transferred; (2)NCDCF_U and NCDCF_I [1], the early multi-behavior based methods which integrate multiple types of behaviors into recommendation by the user-based and item-based neighborhood method, respectively; (3)CMF [7], the state-of-the-art multi-behavior based MF method as discussed before.

To perform comprehensive comparison, we conducted experiments on different training sets (80%, 60% and 40%) to test the models' performance under different sparsity cases. For example, for training data 80%, we randomly select 80% of the data from each types of the behaviors for training and the rest for testing. The random selection was carried out 5 times independently, and we report the average results.

3.2 Experimental Results

Performance Comparison. We evaluate the rating prediction performance for book, music and movie using the above constructed training/testing sets. Since the social relation prediction belongs to the task of link prediction, which is different from rating prediction task and unsuitable to be evaluated by RMSE, here we use social relations as a kind of auxiliary behavior and do not do the social relation prediction task. The experimental results using 10 dimensions to represent the latent factors are shown in Table 2. The parameter values of our GLFM are: $\lambda = 0.2$, $K_s = 6$, K_b=4, which are determined by cross-validation.

From Table 2, we can observe that the multi-behavior based MF methods, CMF, is consistently better than the PMF, which demonstrates that integrating information from multiple types of user behaviors is useful for recommendation. However, the two multi-behavior based neighborhood methods, NCDCF_U and NCDCF_I, do not get consistently better results, which may because that our dataset is very sparse and the neighborhood based methods usually fail to

[1] http://www.douban.com

Table 2: Performance Comparison on different sparsity cases

Behavior	Training	PMF	NCDCF_U	NCDCF_I	CMF	GLFM
Book	80%	0.8150	0.8355	0.7976	0.7849	**0.7684**
	60%	0.8329	0.8367	0.8026	0.8011	**0.7878**
	40%	0.8500	0.8394	0.8143	0.8181	**0.8061**
Music	80%	0.7309	0.7826	0.7367	0.7112	**0.6922**
	60%	0.7326	0.7812	0.7376	0.7187	**0.6943**
	40%	0.7639	0.7859	0.7465	0.7411	**0.7111**
Movie	80%	0.7577	0.8941	1.0866	0.7452	**0.7288**
	60%	0.7671	0.8954	1.0920	0.7581	**0.7374**
	40%	0.7955	0.8971	1.1060	0.7790	**0.7542**

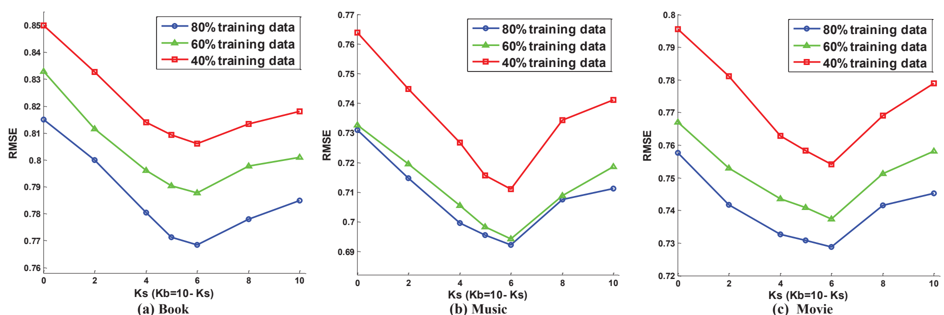

Figure 2: Impact of Parameter K_s (K_b)

find similar neighbors under such sparse data. It is obvious that our GLFM model consistently outperforms other approaches in all sparsity cases, especially achieving significant improvement over CMF, which illustrates that the by modeling the shared and behavior-specific latent factors among behaviors, GLFM can integrate users' multiple types of behaviors into recommendation more effectively.

Impact of Parameter K_s and K_b. In GLFM, the shared latent factors model the correlation between multiple behaviors and the behavior-specific latent factors model the heterogeneity of them. Hence, we investigate the effects of the important parameter in GLFM: the number of shared latent factors K_s and the number of behavior-specific latent factors K_b. In the extreme cases, if $K_s = 0$, it degenerates to PMF, which will not share any information between behaviors; if $K_b = 0$, it degenerates to CMF, which will not model the specific characteristics of different behaviors. Fixing the total number of latent factors ($K_b + K_s$) as 10, Figure 2 shows the performance of GLFM on different training sets with different values of K_s (K_b is also different for $K_b = 10 - K_s$). We can see that, in all cases the RMSE results decrease (prediction accuracy increases) at first with K_s increasing, which demonstrates that sharing information between users' multiple types of behaviors is useful for recommendation; however, when K_s goes greater than a threshold the RMSE increase (prediction accuracy decreases) with K_s increasing (K_b decreasing), which may because that improper information is transferred to harm the recommendation results for lack of modeling the specific characteristics of different behaviors.

4. CONCLUSION

In this paper, we propose a novel recommendation model, GLFM, to integrate multiple types of user behaviors effectively by modeling the correlation and heterogeneity of them. To achieve this goal, GLFM attempts to find a factorization of latent factor space into subspaces that are shared across multiple behaviors and subspaces that are specific to each type of behaviors. Experiment on real-world dataset demonstrate that the proposed method can achieve better recommendation results than other competitors.

5. ACKNOWLEDGEMENTS

This work was supported in part by 973 Program (Grant No. 2010CB327905), National Natural Science Foundation of China (Grant No. 61170127, 61332016).

6. REFERENCES

[1] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-domain mediation in collaborative filtering. In *User Modeling*. 2007.

[2] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.

[3] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, 2012.

[4] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI*, 2009.

[5] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.

[6] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2008.

[7] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD*, 2008.

[8] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *WWW*, 2011.